Hessam S. Sarjoughian

Arizona Center for Integrative Modeling & Simulation
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ 85281, USA

March 2018

Exercise 1: Consider the Map Search Device (MSD) described in Homework Assignment 4. Expand the model as follows:

- If any map updating is interrupted (i.e., True is received on the Green button) during $0 \leq \sigma \leq ta_{updating}(s)$, then the updating will resume immediately after map searching is finished. Updating the map can be interrupted multiple times. For each map updating interruption, the total amount of time allocated to map updating is increased by a finite amount of time $ta_{interrupt}(s)$, for example, 0.3 unit of time.
- Input can arrive on either the input port Green or Red at any time instance.
- Receiving input events on the inGreen input port are ignored while there is on an ongoing map search.
- Only the first input event True received on the inRed input port while updating a map is handled. Input event False received on the inRed input port terminates map updating.
- When input and output events occur simultaneously, output event has priority.

(a) [10 points] Revise the necessary elements of the $M_{Device} = \langle X, S, Y, \delta_{ext}, \delta_{int}, \delta_{conf}, \lambda, ta \rangle$ specification given the above.

(b) [10 points] Implement the M_{Device} such that the amount of time for searching, updating, and interruption can be assigned in the model's constructor. The constructor for this model has three input parameters. Two of these are $ta_{search}(s)$ and $ta_{update}(s)$ for searching and updating maps. The remaining input parameter is $ta_{interrupt}(s)$. Call this model `MSD.java`

Exercise 2: Implement a model capable of creating input trajectories that exercises all behaviors described for M_{Device}. This model can generate only input value True on its outGreen port, but can generate input values True and False on its outRed port. The constructor for this model has two input parameters called $ta_{Green}(s)$ and $ta_{Red}(s)$. The $ta_{Green}(s)$ is the inter-arrival time for map searches and the $ta_{Red}(s)$ for map updates. This model has an input port `Stop`. When the generator receives an input, it immediately stops producing outputs for the `MSD.java`. Call this generator model `Generator.java`. Hint: the Generator model can be a coupled model consisting of two generator models, each dedicated to produced output events for the inGreen and inRed input ports for the MSD.

Exercise 3: Implement a model that can measure the amount of time M_{Device} spends responding to searching and updating maps. The constructor for this model has one input parameter. It is a duration for to observe and gather the following measurements. This model generates a Stop event for the Generator when the observation time $ta_{ObservationTime}(s)$ is reached. Call this transducer model `Transducer.java`.

- Number of successful (completed) map searches. This information is provided using output port `SearchNumber.`
- Number of successful (completed) map updates. This information is provided using output port `UpdateNumber.`
- Average amount of time for successful (completed) map updates. This information is provided using output port `UpdateTime.` The duration for a map update is from the time the request is received until the time the update is completed.

Exercise 4: Implement a model that can be used to study the MSD under for the given experimental settings defined in the Generator and Transducer model. Call this model `EF_MSD.java`. This coupled model consists of the MSD model and an experiment frame called `ExpFrame.java` that is composed of the `Transducer.java` and `Generator.java` simulation models.

Exercise 5: Design and simulate an experiment given the following configuration. Provide the result of this simulation in a tabular form (Tracking log) that has all I/O for all atomic and coupled models without repetition from $t = 0$ to $ta_{ObservationTime}(s)$.

Generator

- $ta_{Green}(s)$ = 1.6, duration between generating input values True events. The first input is generated at $t =$ 1.0.
- $ta_{Red}(s) = 6.7$, duration between generating consecutive input values True and False events. The first input with value True is generated at $t = 5.0$. Every fifth input value is a False event.

MSD

- $ta_{search}(s) = 2.0$, duration for map searching.
- $ta_{update}(s) = 7.0$, duration for map updating. This time duration assumes starting and finishing updating without any interruption.
- $ta_{interrupt}(s) = 0.3$, duration for handling interrupts in map updating phase.

Transducer

- $ta_{ObservationTime}(s) = 80.0$, duration for observation time